CS3002- Information Security

Assignment 1

Natural Language Processing

Name: Mudasir Saeed

Roll no: i21-0592

Section: CS-Z

Introduction:

For this assignment, we created our own unigram, bigram, and trigram language models from scratch. For sentiment classification, we then used a Naive Bayes classifier. The challenge entailed analyzing a dataset of movie reviews, classifying the sentiment of the reviews as "positive" or "negative" using the Naive Bayes approach, and utilizing the language models to predict the next word in a phrase.

1. ## Data Loading and Preprocessing:

   For this assignment, we created our own unigram, bigram, and trigram language models from scratch. For sentiment classification, we then used a Naive Bayes classifier. Processes such as language models to anticipate the next word in a phrase and classify the attitude of movie reviews as "positive" or "negative" were part of the challenge.We used a CSV-formatted dataset including movie reviews. Using Python's csv package, the reviews and the associated feelings (positive/negative) were loaded. After that, text preprocessing techniques were used to clean and tokenize the data. Change to lowercase. Take off the HTML tags (
). Take out the punctuation Convert tokens into language. By taking these actions, the data was guaranteed to be in the right format for tasks involving language modeling and categorization.The naive Bayes method.

2. Unigram Model:

Based on the frequency of occurrence of each word in the corpus, the unigram model computed its probability. According to this concept, every word exists independently of the others.

3. Bigram Model:

Word pairings are created by the bigram model, which represents the likelihood of one word coming after another. This enables the model to forecast, given the present word, the word that will appear next in a sequence.

4. ## Trigram Model:

The trigram model uses the preceding two words to predict the following word. This improves upon the bigram model by providing additional context for the prediction process.

5. Word Prediction:

Based on the provided context (prior word(s)), we created prediction functions for bigram, trigram, and unigram models to forecast the following word.

Unigram: Forecasts using the frequency of individual words.
Bigram: Forecasts using word pairings, or the word of the moment.
Trigram: Uses the previous two words to predict the following word.

6. Naive Bayes Theorem:

Based on the preprocessed reviews, we deployed a Naive Bayes classifier for sentiment categorization. The classifier operates as follows:
determining the feelings' prior probabilities (positive/negative).
estimating the chances of each word in the review based on the sentiment category.
categorizing a fresh review according on the likelihoods and product of the previous reviews.

7. Generated Sentence classification:

The Naive Bayes classifier receives the sentence produced by the trigram model and uses it to determine whether it is "positive" or "negative."

Conclusion:

We successfully created and trained trigram, bigram, and unigram models from scratch through this assignment. Next, we trained a Naive Bayes sentiment classifier. We investigated the entire cycle of language modeling and classification by creating sentences using the trigram model and categorizing their sentiment. By balancing priors and smoothing likelihoods, the classifier's initial bias was reduced, resulting in more reliable performance.